

**AMENDMENTS TO THE CLAIMS**

1. (Currently Amended) A client-server system comprising a thin client interface residing on at least one client and an object manager and an application residing on one or more servers, said object manager interposed between said client and said application server, said object manager being operable to handle requests from multiple thin clients by maintaining a status of each client in a corresponding object manager thread, and said application server comprising one or more of business objects, and business components.
2. (Original) The client-server system of claim 1 wherein the application server comprises a database server.
3. (Currently Amended) The client-server system of claim 1 ~~wherein~~ further comprising object manager run-time engines that operate on the business objects and business components.
4. (Original) The client-server system of claim 3 wherein the business objects and business components comprise applets and application objects.
5. (Original) The client-server system of claim 3 wherein object manager run time engines enforce repository-defined business processes and rules.
6. (Currently Amended) The client-server system of claim 1 ~~having~~ comprising application objects executing on the client.
7. (Currently Amended) The client-server system of claim 1 ~~having~~ comprising user interface objects executing on the client.
8. (Original) The client-server system of claim 1 comprising session-based network protocols connecting the client to the object manager.

9. (Currently Amended) A method of connecting a client and one or more servers in a client server network, wherein said client is a thin client, and said one or more servers comprise an object manager and an application residing on one or more servers, said object manager interposed between said client and said application server, and said application server comprising one or more of business objects, and business components;  
instantiating said one or more business objects; and  
establishing a session based network connection between the thin client and the one or more servers, wherein said object manager being operable to handle requests from multiple thin clients by maintaining a status of each client in a corresponding object manager thread.
10. (Original) The method of claim 9 comprising instantiating object manager run-time engines to operate on the business objects and business components.
11. (Original) The method of claim 9 wherein the business objects and business components comprise applets and application objects.
12. (Original) The method of claim 9 wherein object manager run time engines enforce repository-defined business processes and rules.
13. (Original) The method of claim 9 wherein application objects execute on the client.
14. (Original) The method of claim 9 wherein the user interface objects execute on the client.
15. (Previously Presented) A client server system comprising:  
a thin client interface being operable to execute on a client computer;  
an object manager being operable to reside on one or more servers, being operable to handle requests from multiple thin clients by maintaining a status of each client in a corresponding object manager thread; and

an application being operable to reside on one or more servers, the application server comprising one or more business objects, wherein the object manager is interposed between the thin client and the application server.

16. (Previously Presented) The system of claim 15 wherein the object manager is a multi-tasking, multi-thread process.
17. (Previously Presented) The system of claim 15 wherein the thin client interface being further operable to provide persistent sessions between the client and application server.
18. (Previously Presented) The system of claim 15 wherein the business object is a horizontal application.
19. (Previously Presented) The system of claim 15 wherein the business object is a vertical application.
20. (Previously Presented) The system of claim 15 wherein the business object is an internet application.